



# S-TRADE

## Group 30

KWAN Chak Lam 20095398  
Deng Ken 20400660  
Hou Jiefeng 20361723

Group Members:  
[clkwanaa@connect.ust.hk](mailto:clkwanaa@connect.ust.hk)  
[kdengab@connect.ust.hk](mailto:kdengab@connect.ust.hk)  
[jhouad@connect.ust.hk](mailto:jhouad@connect.ust.hk)

## Table of content

|   |              |
|---|--------------|
| <b>1. Introduction .....</b>                                | <b>2</b>     |
| <b>2. Design and Implementation .....</b>                   | <b>3-17</b>  |
| <b>2.1 Requirements Analysis .....</b>                      | <b>3-5</b>   |
| <b>2.2 Design .....</b>                                     | <b>6-17</b>  |
| <b>2.3 Implementation .....</b>                             | <b>17-21</b> |
| <b>3. Testing and Evaluation .....</b>                      | <b>22-26</b> |
| <b>4. Unfinished functions and future enhancement .....</b> | <b>26-27</b> |
| <b>4.1 Unfinished functions .....</b>                       | <b>26</b>    |
| <b>4.2 Future enhancement .....</b>                         | <b>27</b>    |
| <b>5. Difficulties encountered .....</b>                    | <b>28</b>    |
| <b>6. Conclusion.....</b>                                   | <b>28</b>    |
| <b>7. Reference .....</b>                                   | <b>28</b>    |

## **1. Introduction**

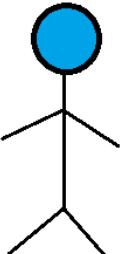
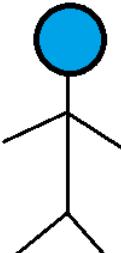
In this decade, environmental protection is becoming a concern of human beings. The 3R concept, which means “reuse”, “recycle” and “reduce”, were also been introduced to citizen. Therefore, there is a needed of platforms for exchanging unused or second-hand items to encourage people reusing objects and reducing waste. Especially in University, teenagers like buying things and pursing the digital trend. Many outdated but still usable items were thrown away. Moreover, some students buy the textbook for studying but it may become useless for them after passing the course. Hence, the idea of S-Trade is being introduced. S-Trade is a platform form HKUST student to trade or selling their items.

In this report, a overview of S-Trade will be introduced. At first, the design idea of S-Trade will be mentioned, with assist of UML diagrams such as user case model. Moreover, this report included all finished features and new features which are added after the first presentation. The technical details of how to develop some of the features will also be mentioned. UI test was also implemented on S-Trade. The result is also attached in the report. The unfinished functions, future enhancement and difficulties we faced will be at the last part of the report.

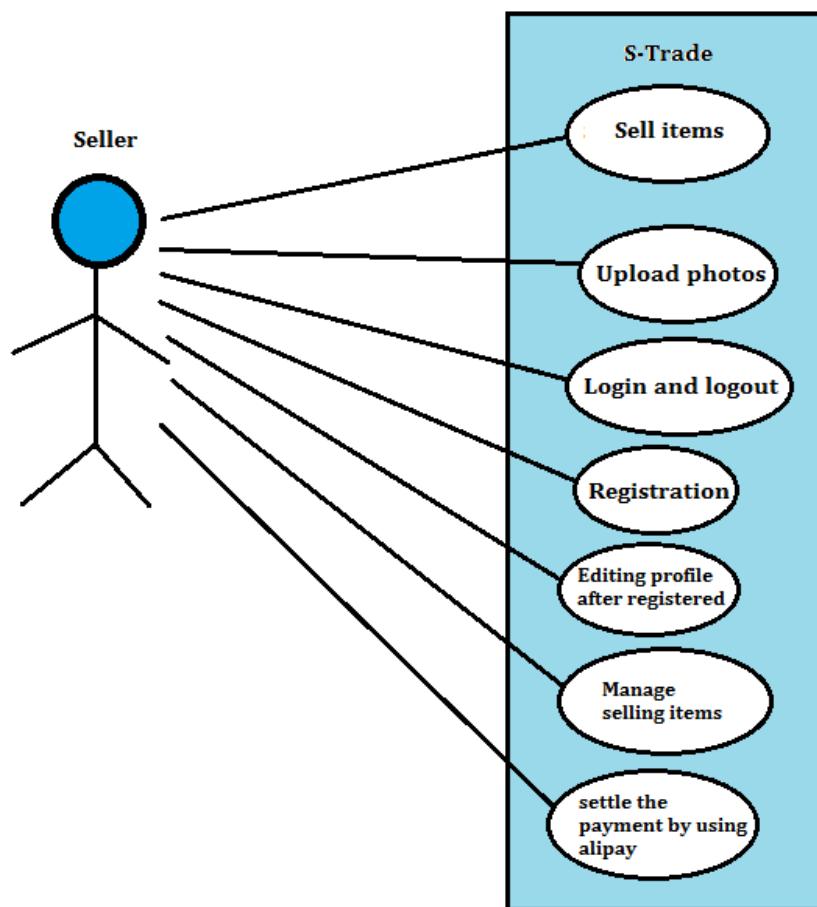
## 2. Design and Implementation

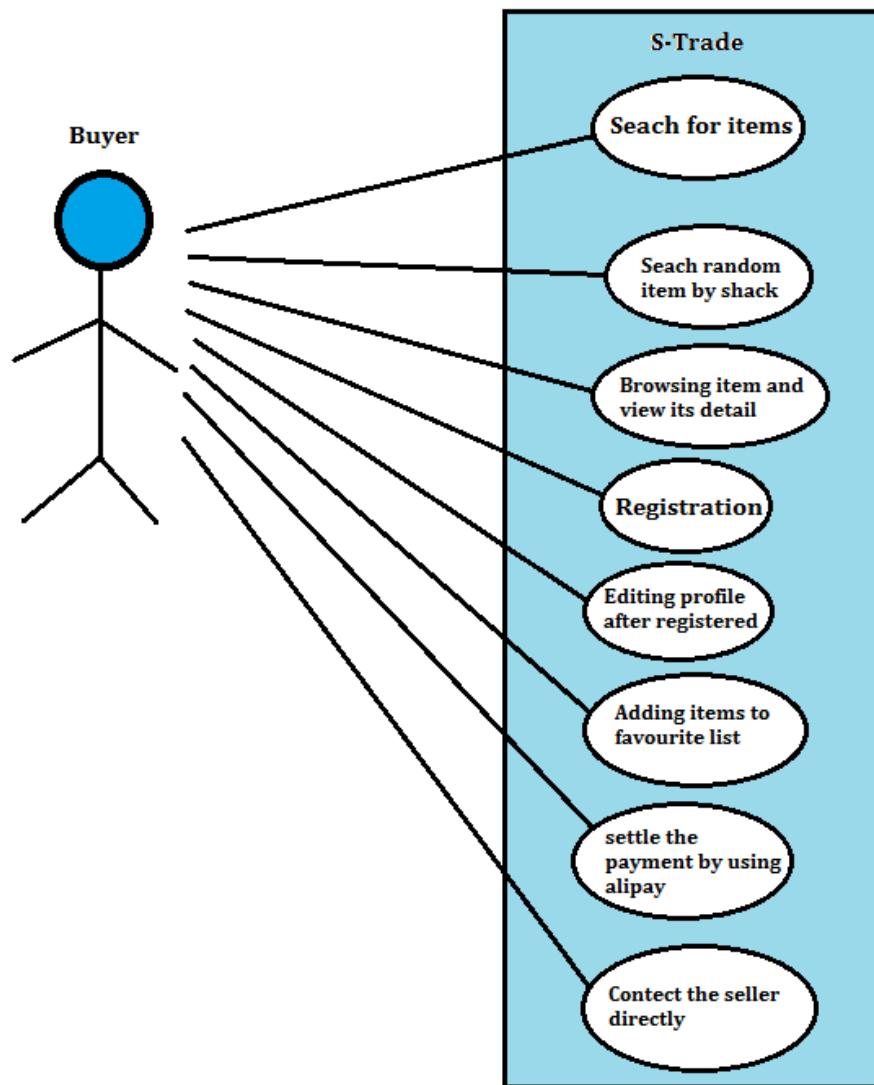
### 2.1 Requirements Analysis

#### Major actor of the application

|   |   |
|---|---|
| <br>Seller | User: Seller<br><br>Seller is the user who use the application to sell their item. Seller can be any individual registered user form HKUST. The user do not required any experience on using application on smart phone and selling item on applications because the application because S-Trade will provide instruction step by step.   |
| <br>Buyer | User: Buyer<br><br>Buyers are the user who use the application to look for second-hand item. Buyer can be any individual user form HKUST. Difference from seller, buyer is not required to register. The user do not required any experience on using application on smart phone and buying item on applications because the application because S-Trade will provide instruction step by step. |

#### Use case model





## 2.2 Design

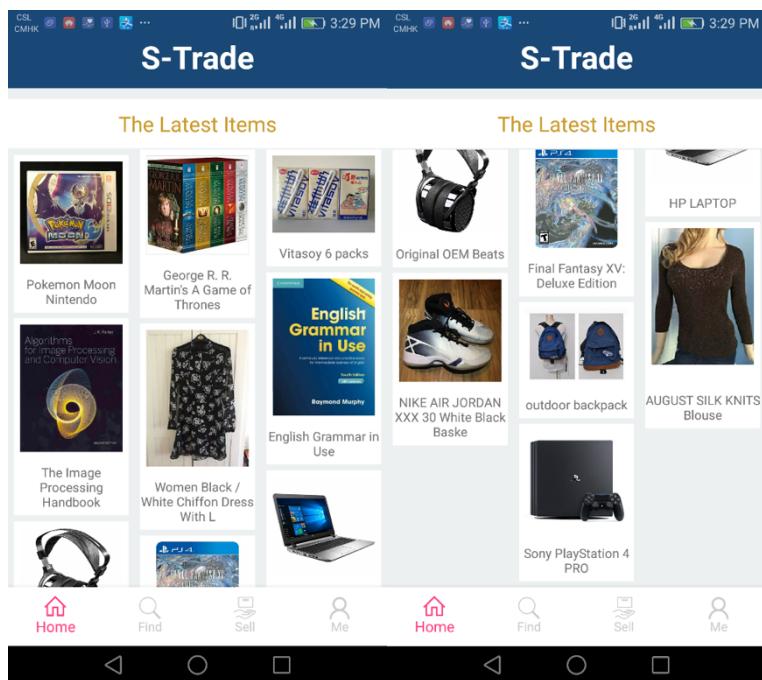
### MainActivity

The main activity consists of five fragments which are controlled by bottom navigation bar.

| Main Component        | Description   |
|-----------------------|---|
| Bottom navigation bar | Users can go to different view directly with tapping on a bottom navigation icon.   |
| Sharedpreferences     | Get log-in state data from preferences file, if data is empty, HomeFragment, FindFragment, SellFragment and UserFragment will be displayed in MainActivity. If data is not empty, LogInFragment will instead of UserFragment in MainActivity. |

Below is a brief description of fragments:

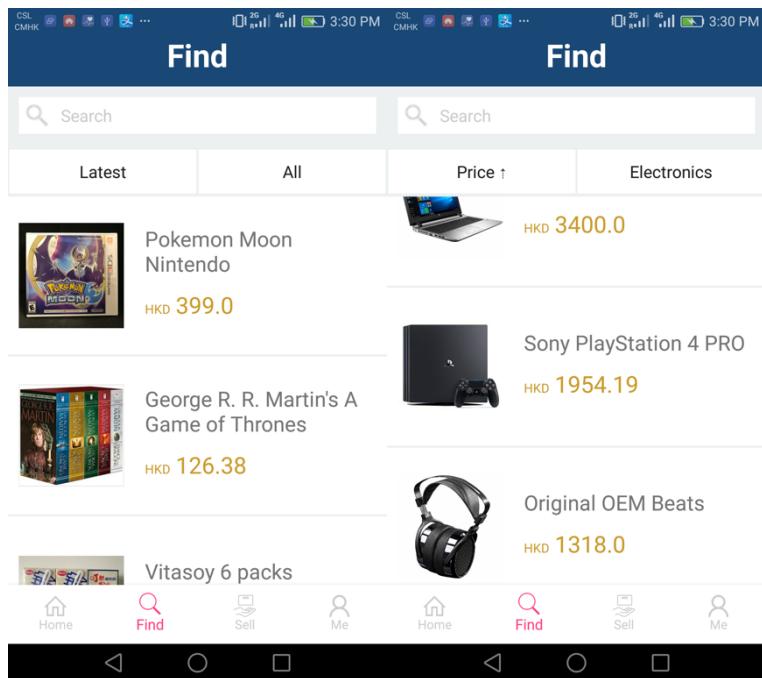
#### HomeFragment



The home fragment shows the latest items.

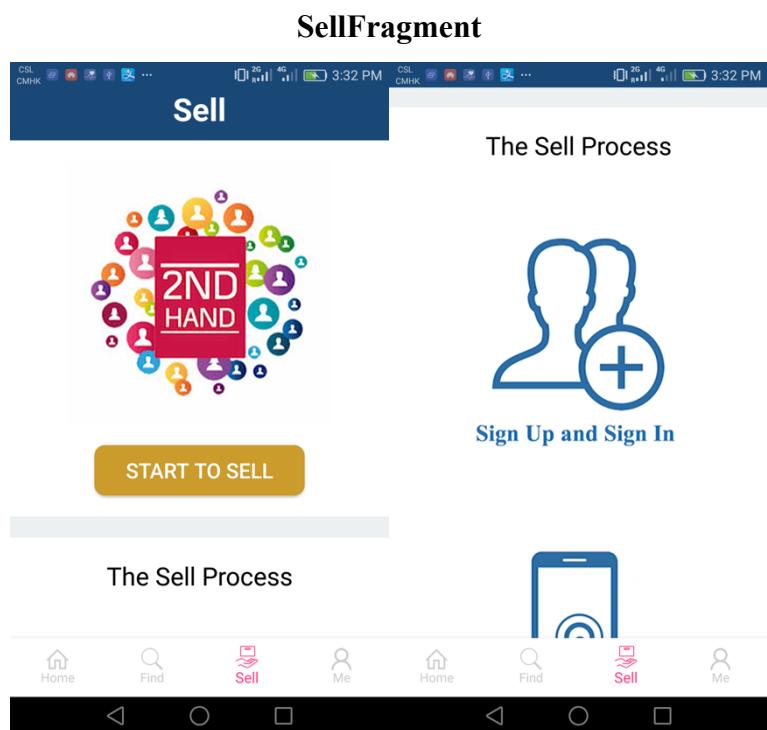
| Main Component | Description                                   |
|----------------|---|
| RecyclerView   | Using staggered grid layout to display items. |

## FindFragment



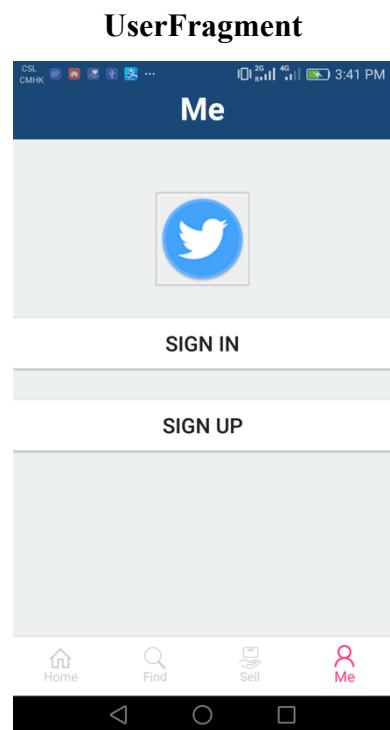
The find fragment shows a list of items with search function, sorting function and selecting category function.

| Main Component | Description  |
|----------------|--|
| SearchView     | Click search bar to start SearchActivity.  |
| Spinner        | Click sort selection box to select items sorting method, and click category selection box to select a category of items. |
| ListView       | Using ListView to display items.   |



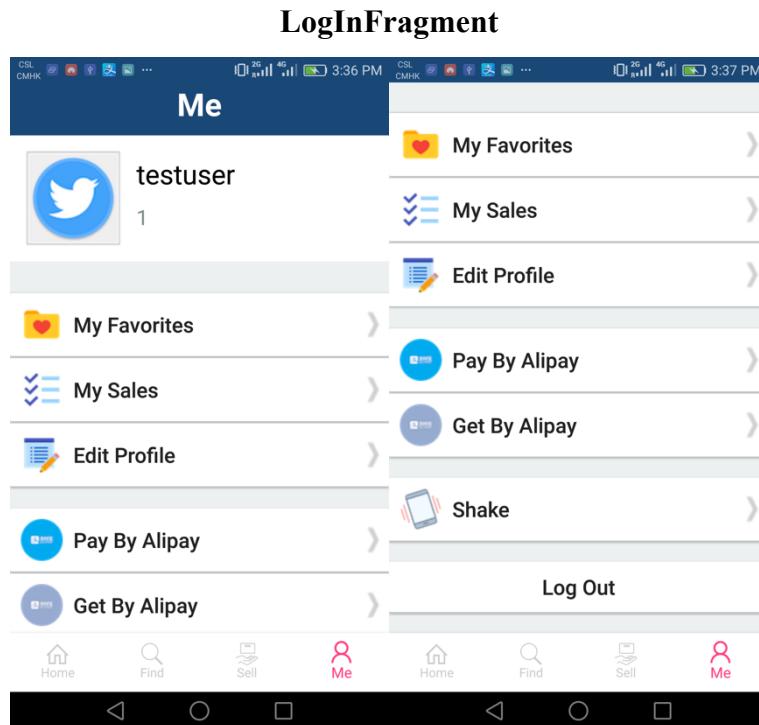
The sell fragment is for sellers to start to sell a new item.

| Main Component | Description  |
|----------------|--|
| Button         | Click “start to sell” button to start NewItemActivity. |



The user fragment is for users to sign in or sign up.

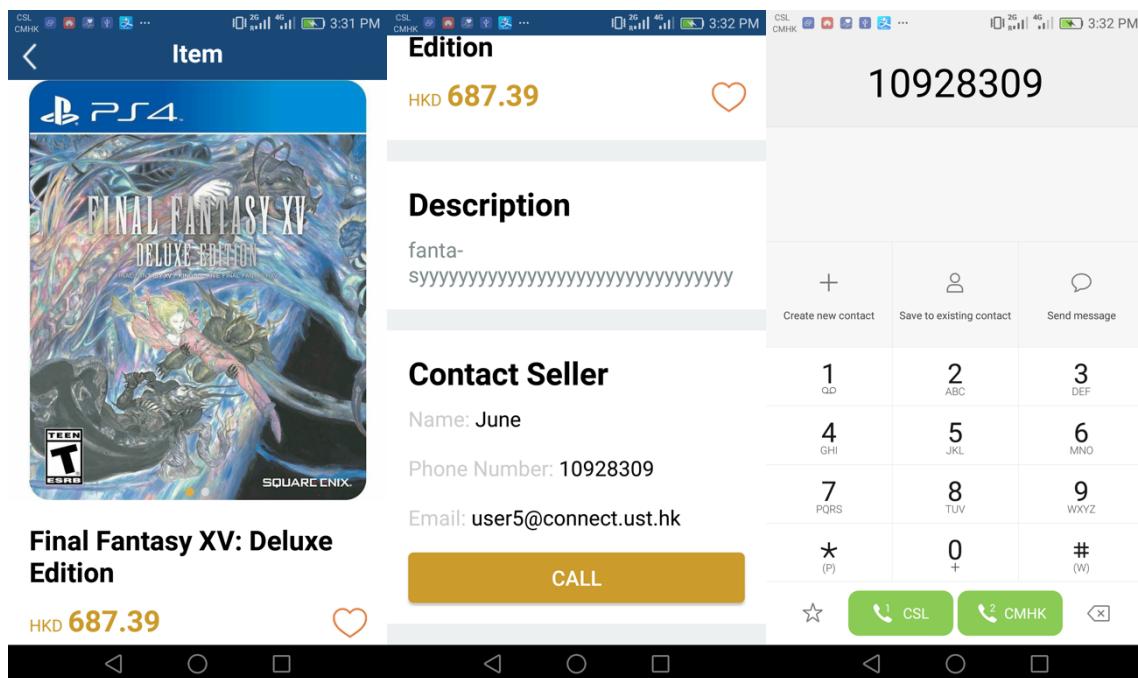
| Main Component | Description  |
|----------------|--|
| Button         | Click “sign in” button to start SignInActivity, and click “sign up” to start SignUpActivity. |



The log in fragment only shows after user signs in.

| Main Component    | Description   |
|-------------------|---|
| Button            | <ol style="list-style-type: none"> <li>1. Click “my favorite” button to start FavoriteActivity.</li> <li>2. Click “my sale” button to start SaleActivity.</li> <li>3. Click “edit profile” button to start ProfileActivity.</li> <li>4. Click “pay by alipay” button to access to alipay application.</li> <li>5. Click “get by alipay” button to access to alipay application.</li> <li>6. Click “shake” button to start ShakeActivity.</li> <li>7. Click “log out” button to back to UserFragment (using UserFragment instead of LogInFragment).</li> </ol> |
| Sharedpreferences | Clear log-in state data which storing at preferences file after user log out.   |

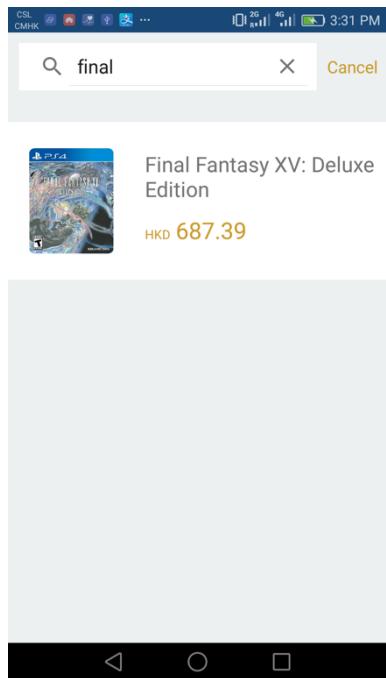
## ItemActivity



The item activity shows a particular item detail.

| Main Component | Description  |
|----------------|--|
| RollPagerView  | Using RollPagerView to auto play the photos of item.   |
| ImageView      | Click heart image to add item to user's favorite list. |
| Button         | Click "call" button to go to phone dialing interface.  |

## SearchActivity



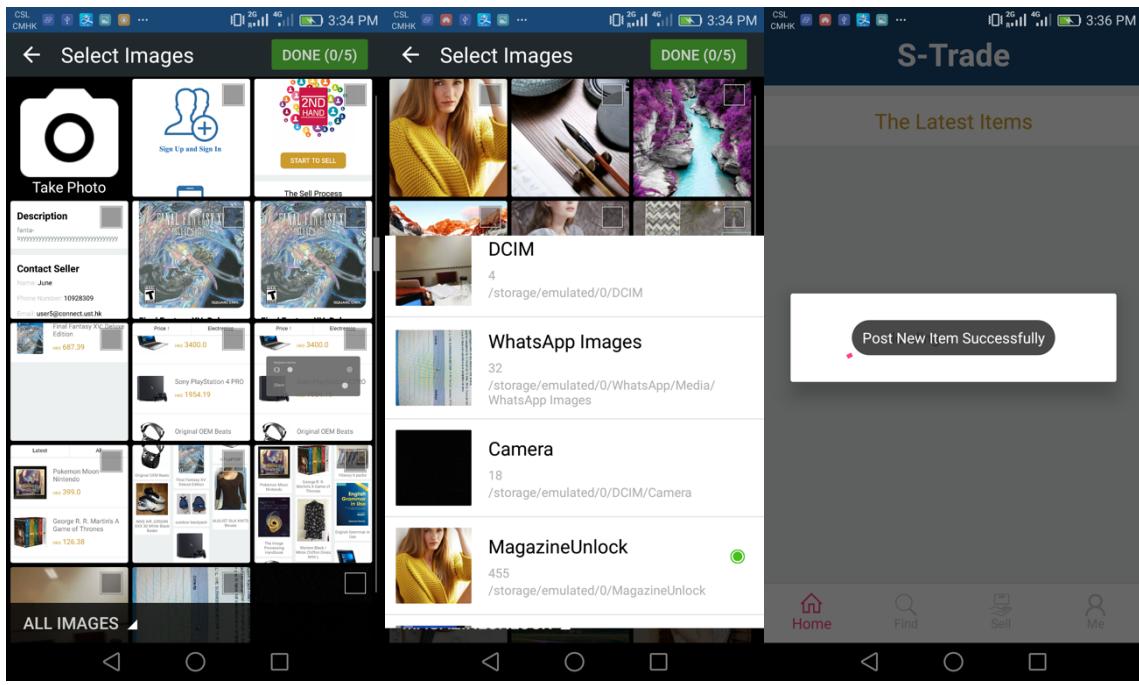
The search activity is for users to search items.

| Main Component | Description   |
|----------------|---|
| SearchView     | Using SearchView to create a search bar, so user can input key words and find relevant items. |
| ListView       | Using ListView to display the search results.   |

### NewItemActivity

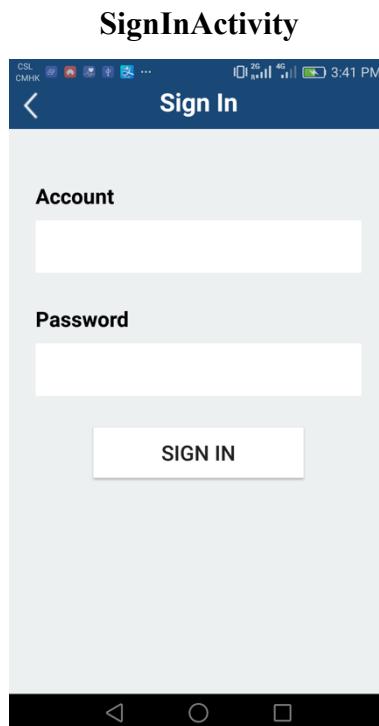
This screenshot shows the "NewItemActivity" interface. The screen is divided into several sections:

- Item Detail:** Includes fields for "Item Name" (steelseries mouse), "Price" (276.5), "Description" (brand new), "Category" (Electronics), and "Price" (276.5).
- Contact Information:** Includes fields for "Contact Name" (Sam), "Phone Number" (93028300), "Email" (user1@connect.ust.hk), and a large yellow "SELL IT" button.
- Item Photo:** A section with an "UPLOAD" button.
- Category:** A dropdown menu currently set to "Electronics".



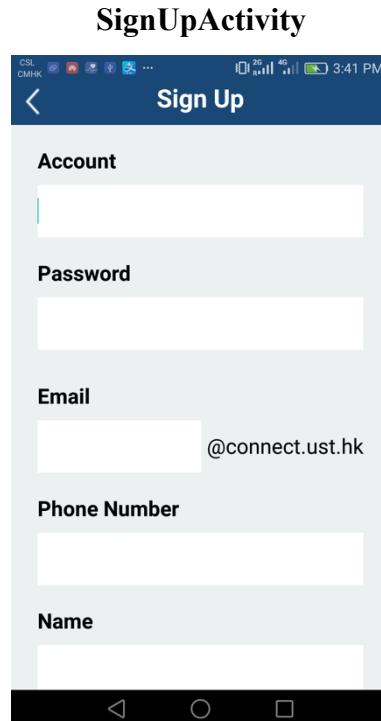
The new item activity is for users to input all required information of selling item. Also, the contact information will load from the server at first.

| Main Component | Description  |
|----------------|--|
| EditView       | Using EditView for users to input required information.  |
| Button         | 1. Click “upload” button to upload photo from camera or gallery.<br>2. Click “sell” button to upload selling item to server. |



The sign in activity is for user to log in.

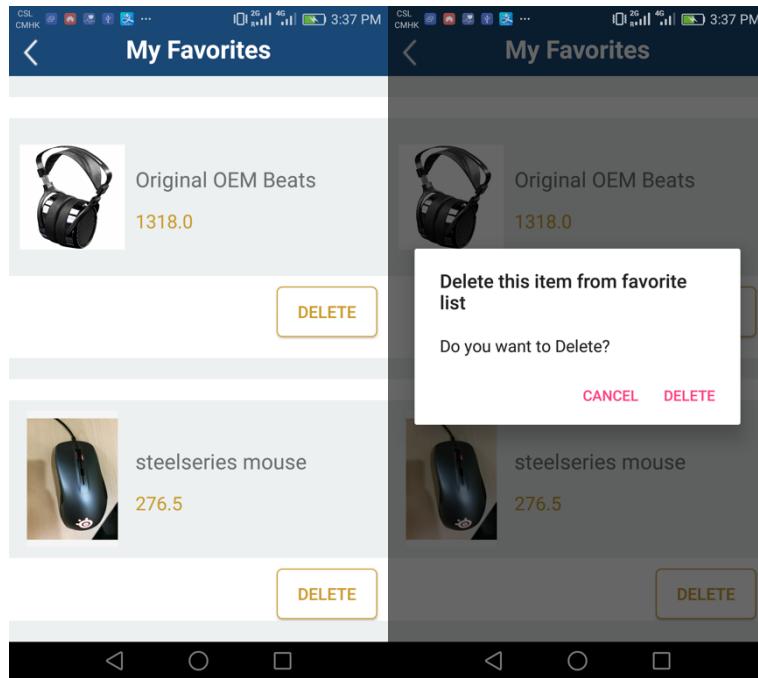
| Main Component    | Description   |
|-------------------|---|
| Button            | Click “sign in” button (if log-in is successful) to back to MainActivity. |
| Sharedpreferences | Using Sharedpreferences to store log-in state.                            |



The sign-up activity is for user to create a S-trade account.

| Main Component | Description   |
|----------------|---|
| Button         | Click “sign up” button (if sign-up is successful) to back to MainActivity. Also, the account information will send to server. |

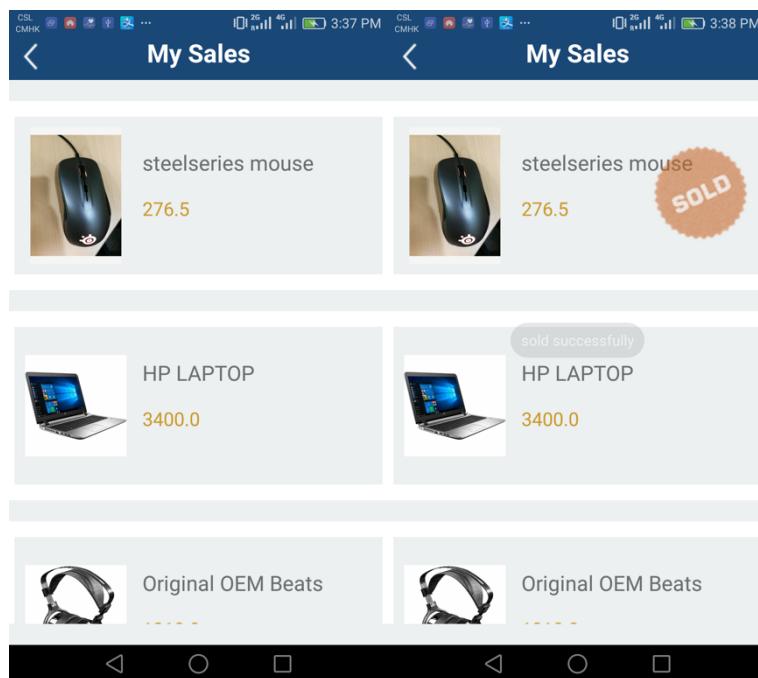
## FavoriteActivity



The favorite activity shows a list of user's favorite items which added before.

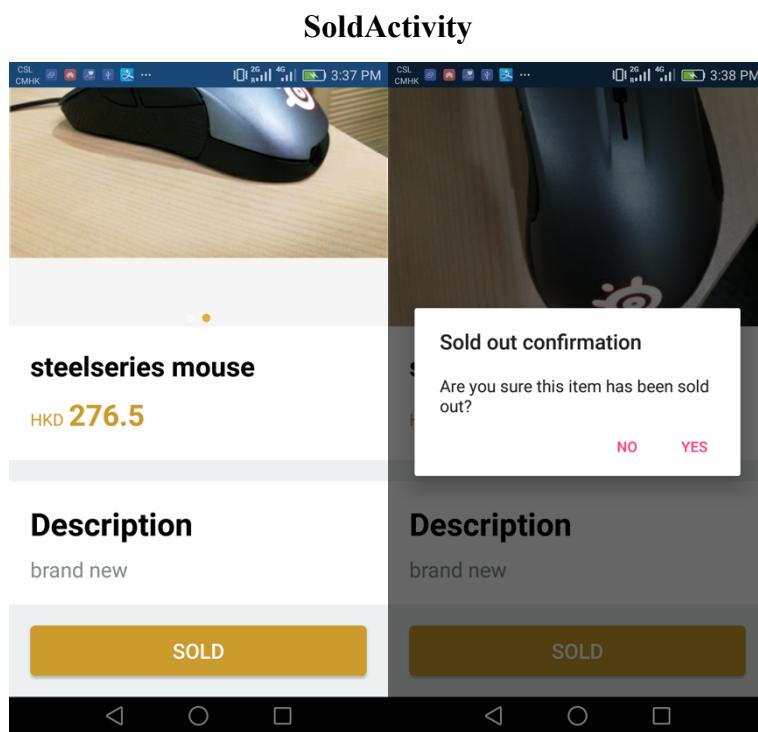
| Main Component | Description   |
|----------------|---|
| ListView       | Using ListView to display a list of favorite items.   |
| Button         | Click "delete" button to delete a particular item from user's favorite list. Also, a dialog will pop up to let user confirm deleting command. |

## SaleActivity



The sale activity shows a list of user's selling items.

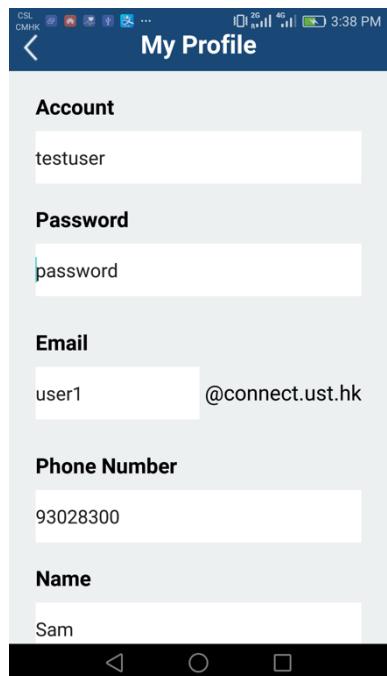
| Main Component | Description   |
|----------------|---|
| ListView       | Using ListView to display a list of user's selling items. And user can click one particular item to start SoldActivity. In addition, if a particular item has been sold out, a sold logo will be displayed. |



The sold activity is for user to change the state of selling item to “sold”.

| Main Component | Description  |
|----------------|--|
| Button         | Click “sold” button to change state of item on server and back to LogInFragment. Also, a dialog will pop up to let user to confirm changing state command. |

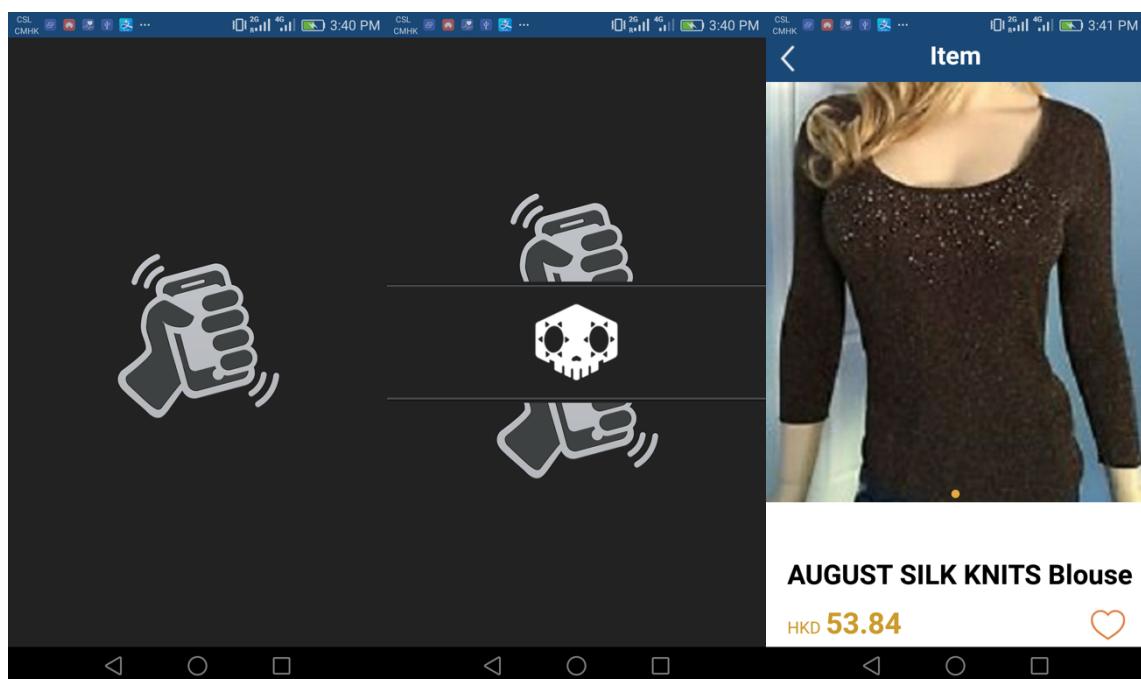
## ProfileActivity



The profile activity is for user to edit profile information.

| Main Component | Description  |
|----------------|--|
| Button         | Click “save” button to save new profile information to server. |

## ShakeActivity



The shake activity is for user to shake phone to get a random item.

| Main Component | Description |
|----------------|-------------|
|                |             |

|        |   |
|--------|---|
| Sensor | Sensor of the phone. To sense any physical movement of the phone. A parameter can be set to how sensitive the sensor will be. |
|--------|---|

## 2.3 Implementation

### Data stored in client

Most of data will be stored on a database server, the only parameter stored in client is *USERID*. After a user signed in, the *USERID*, which is the unique identifier of user, will be stored in the local cache. We use interface SharedPreferences to store *USERID*, the example how we store *USERID* is as below:

```
public void storeLogInData(int i) {
    SharedPreferences sharedpreferences = getSharedPreferences("logInState", Context.MODE_PRIVATE);
    SharedPreferences.Editor editor = sharedpreferences.edit();

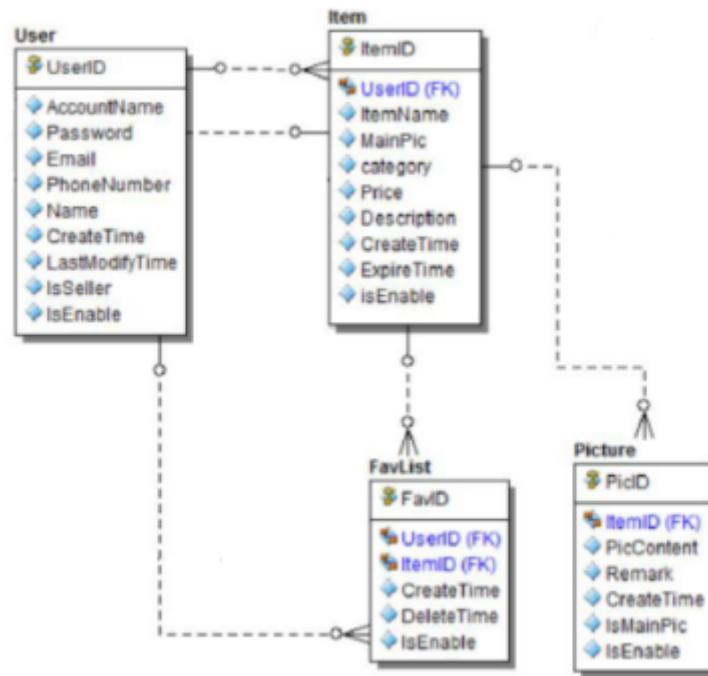
    editor.putString("userName", userName.getText().toString());
    editor.putString("userID", String.valueOf(i));
    editor.putInt("loginState", 0);
    editor.commit();
}
```

The example how we get the *USERID* is as below:

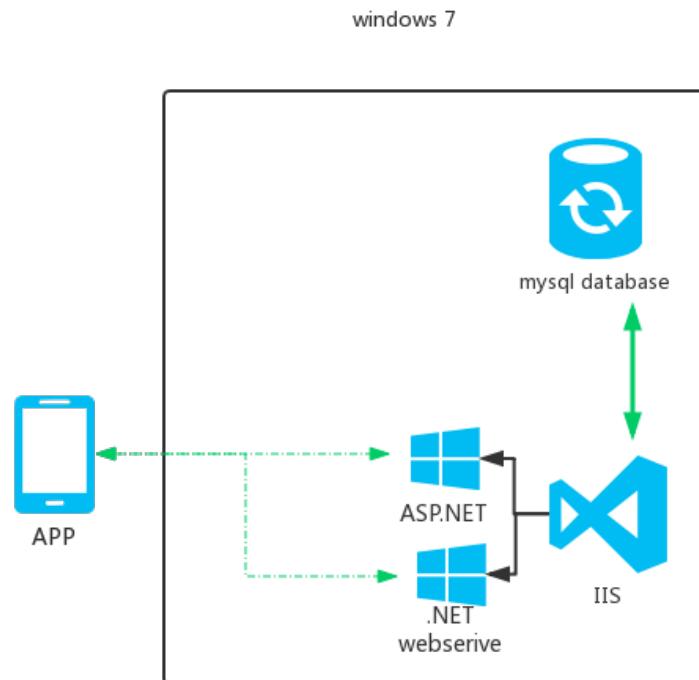
```
SharedPreferences sharedpreferences = getSharedPreferences("logInState", Context.MODE_PRIVATE);
final String uid=sharedpreferences.getString("userID","");
usid=uid;
```

### Data stored in server

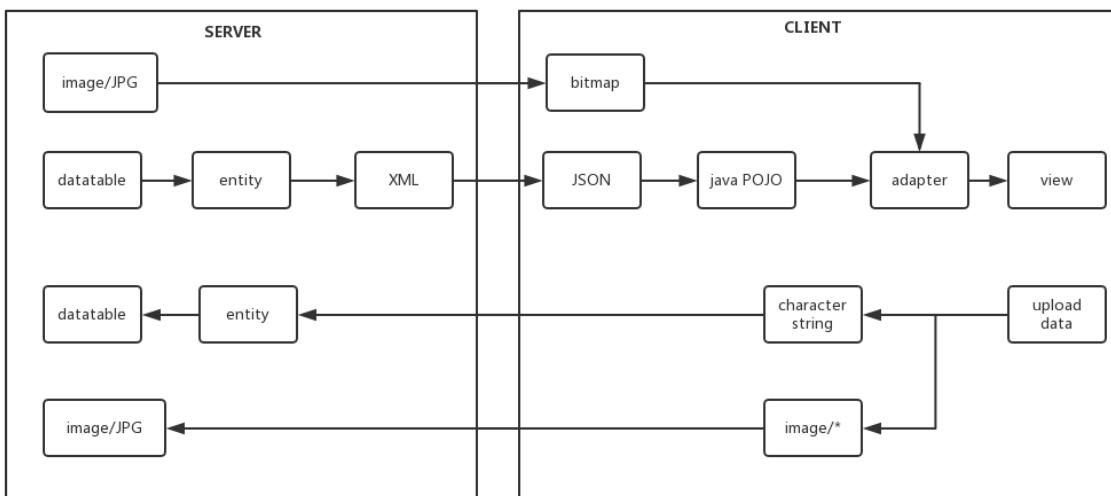
Database E-R diagram is as below:



Connection between database and website is as below:



the concept of the how data transmit between server and client is as below:



For client, there are totally fourteen web methods to get the entity from server. Every activity(fragment) will establish an asynctask to call the web method. The example of code is as below:

```

private class MyTask extends AsyncTask<String, Integer, List<item>> {

    ProgressDialog dialog = new ProgressDialog(getActivity());

    @Override
    protected void onPreExecute() { ... }

    @Override
    protected List<item> doInBackground(String... params) {
        try {
            List<item> items = getinfo.getitemlistall();
            bms.clear();
            for (int i = 0; i < items.size(); i++) {
                Bitmap bm = httphelper.getimage(String.valueOf(items.get(i).getItemid()));
                bms.add(bm);
            }
            return items;
        } catch (Exception e) {
            return null;
        }
    }
}

```

After this thread successfully get the entity, to fulfill the adapter and then the view of the activity will show the item. The example is as below:

```

@Override
protected void onPostExecute(final List<item> result) {
    initdata(result);
    recyclerView.setLayoutManager(new StaggeredGridLayoutManager(3, StaggeredGridLayoutManager.VERTICAL));

    adapter = new MasonryAdapter(getContext(), productList);
    recyclerView.setAdapter(adapter);

    SpacesItemDecoration decoration = new SpacesItemDecoration(16);
    recyclerView.addItemDecoration(decoration);
    adapter.setOnItemClickListener((position, product) -> {
        Intent intent = new Intent();
        intent.putExtra("itemID", String.valueOf(result.get(position).getItemid()));
        intent.setClass(getActivity(), ItemActivity.class);
        startActivity(intent);
    });
}

dialog.dismiss();
}

```

The main features of this app are graphic and sensor. SHAKE activity user sensor to implement the function which includes animation and audio. First, override the onstart() method in the shakeactivity. open the sensor and register the listener.

```

@Override
protected void onStart() {
    super.onStart();
    mSensorManager = ((SensorManager) getSystemService(SENSOR_SERVICE));
    if (mSensorManager != null) {
        mAccelerometerSensor = mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
        if (mAccelerometerSensor != null) {
            mSensorManager.registerListener(ShakeActivity.this, mAccelerometerSensor, SensorManager.SENSOR_DELAY_UI);
        }
    }
}

```

Additionally, onpause() method must be overrided because sensor will not be turn off automatically if user turn into any other activity. unregister the sensor in onpause() method:

```

@Override
protected void onPause() {
    if (mSensorManager != null) {
        mSensorManager.unregisterListener(ShakeActivity.this);
    }
    super.onPause();
}

```

Then, override the onsensorchanged() method, monitor the sensor until user shakes the phone. Then start animation.

Finally, after the animation ends, get the random index of the item from the server, use intent to start the item activity.

```
@Override
public void handleMessage(Message msg) {
    super.handleMessage(msg);
    switch (msg.what) {
        case START_SHAKE:
            //This method requires the caller to hold the permission VIBRATE.
            mActivity.mVibrator.vibrate(300);

            mActivity.mSoundPool.play(mActivity.mWeiChatAudio, 1, 1, 0, 0, 1);
            mActivity.mTopLine.setVisibility(View.VISIBLE);
            mActivity.mBottomLine.setVisibility(View.VISIBLE);
            mActivity.startAnimation(false);
            break;
        case AGAIN_SHAKE:
            mActivity.mVibrator.vibrate(300);
            break;
        case END_SHAKE:
            mActivity.isShake = false;
            mActivity.startAnimation(true);
            break;
        case GO_NEXT:
            Intent intent = new Intent();
            intent.putExtra("itemID", String.valueOf(itemid));
            intent.setClass(mActivity, ItemActivity.class);
            mActivity.startActivity(intent);
            break;
    }
}
```

### 3. Testing and Evaluation

We used direct user testing as our testing method. Here is the test result on different activities:

| ID    | Input                           | Expected result                                     | Actual result                                       | Result |
|-------|---------------------------------|---|---|--------|
| STA 1 | Start S-Trade                   | S-Trade is started                                  | S-Trade is started                                  | Yes    |
| STA 2 | Wait for application initialize | Main activity and recyclerview of items will appear | Main activity and recyclerview of items will appear | Yes    |

Initialize Application

| ID | Input  | Expected result                                       | Actual result   | Result |
|----|--|---|---|--------|
| M1 | Click on any item                                | Details of the corresponding item will be displayed   | Details of the corresponding item will be displayed   | Yes    |
| M1 | Item with name and at least one photo            | Item photo and name will be displayed by recyclerview | Item photo and name will be displayed by recyclerview | Yes    |
| M3 | Press "Find" button on the bottom navigation bar | Will switch to search page                            | Will switch to search page                            | Yes    |
| M4 | Press "Sell" button on the bottom navigation bar | Will switch to sell page                              | Will switch to sell page                              | Yes    |
| M5 | Press "Me" button on the bottom navigation bar   | Will switch to my profile                             | Will switch to my profile                             | Yes    |

Main Activity

| ID | Input   | Expected result  | Actual result  | Result |
|----|---|--|--|--------|
| S1 | Click the seach bar   | Keboard will appear for input keyword                  | Keboard will appear for input keyword                  | Yes    |
| S2 | Type in any keywords which has related items name in the database and click search button | Items name which contain the keyword will be displayed | Items name which contain the keyword will be displayed | Yes    |
| S3 | Type in any keywords do not have related item in the database                             | No result will be displayed                            | No result will be displayed                            | Yes    |
| S4 | Click the "Latest" button   | A dropdown list with                                   | A dropdown list with                                   | Yes    |

|    |   |  |  |     |
|----|---|--|--|-----|
|    |   | different sorting criteria will pop out                | different sorting criteria will pop out                |     |
| S5 | Click the "All" button  | A dropdown list with different categories will pop out | A dropdown list with different categories will pop out | Yes |
| S6 | Click any sorting criteria in the Latest button dropdown list | All displaying items will be sorted by that criteria   | All items displayed will be sorted by that criteria    | Yes |
| S7 | Click any category in the All button dropdown list            | All items under that category will be listed out       | All items under that category will be listed out       | Yes |
| S8 | Click on any result item                                      | Detail of the item will be displayed                   | Detail of the item will be displayed                   | Yes |

Search

| ID  | Input   | Expected result   | Actual result   | Result |
|-----|---|---|---|--------|
| SU1 | Click the "Sell" button after logged in   | Activity for uploading new items appear and the contact information table will automatically fill by user's personal info.        | Activity for uploading new items appear and the contact information table will automatically fill by user's personal info.        | Yes    |
| SU2 | Click on any box in the upload new item page  | Keyboard will appear for input  | Keyboard will appear for input  | Yes    |
| SU3 | Click the "Upload" button under the Item photo  | Page for selecting image will appear  | Page for selecting image will appear  | Yes    |
| SU4 | Click the "take photo" button in the select image page                                      | Change to camera for taking photo   | Change to camera for taking photo   | Yes    |
| SU5 | Select few photos and click the "Done" button on the right head corner                      | A message "upload new item page "totally x photo is selected" will appear and the screen will switch back to upload new item page | A message "upload new item page "totally x photo is selected" will appear and the screen will switch back to upload new item page | Yes    |
| SU6 | Click the category dropdown list  | A dropdown list with different category for selection will appear   | A dropdown list with different category for selection will appear   | Yes    |
| SU7 | Select any category in dropdown list  | The dropdown list will roll back  | The dropdown list will roll back  | Yes    |
| SU8 | Fill in the information about the followed the instruction and select at least one photo in | A message indicates the upload is success will appear   | A message indicates the upload is success will appear   | Yes    |

|      |  |  |  |     |
|------|--|--|--|-----|
|      | upload new item page and click sell it button                            |  |  |     |
| SU9  | Have not log in and click the "Sell" button                              | An error message appear which ask the user log in      | An error message appear which ask the user log in      | Yes |
| SU10 | Empty anyone of the box in upload new item page and click sell it button | An error message warn that all the box cannot be empty | An error message warn that all the box cannot be empty | Yes |

Sell and upload new item

| ID  | Input  | Expected result   | Actual result   | Result |
|-----|--|---|---|--------|
| P1  | Click the "Sign in" button   | Redirect to sign in page  | Redirect to sign in page  | Yes    |
| P2  | Click the "Sign up" button   | Redirect to sign up page  | Redirect to sign up page  | Yes    |
| P3  | Fill in correct user info. in sign in page and click "sign in"                         | Redirect to the user profile page and logged in                         | Redirect to the user profile page and logged in                         | Yes    |
| P4  | Fill in incorrect user info. or empty the username or password box and click "sign in" | An error message will appear  | An error message will appear  | Yes    |
| P5  | Fill in all the user info. in sign up page and click "sign up"                         | A message telling the user is successfully register will pop out        | A message telling the user is successfully register will pop out        | Yes    |
| P6  | Empty any box in the sign-up page and click "sign up"                                  | An error message will appear  | An error message will appear  | Yes    |
| P7  | Click the "My favourite" button  | Redirect to my favourite page and all the marked item will be displayed | Redirect to my favourite page and all the marked item will be displayed | Yes    |
| P8  | Click on any item in my favourite page   | Details of the corresponding item will be displayed                     | Details of the corresponding item will be displayed                     | Yes    |
| P9  | Click the "Delete" button on any item in my favourite page                             | A confirm message will appear   | A confirm message will appear   | Yes    |
| P10 | Click delete for the confirm message   | The corresponding item will be removed                                  | The corresponding item will be removed                                  | Yes    |

|     |   |  |  |     |
|-----|---|--|--|-----|
|     |   | from the favourite list of the user  | from the favourite list of the user  |     |
| P11 | Click cancel for the confirm message  | Return to the item page  | Return to the item page  | Yes |
| P12 | Click the "My sales" button   | Redirect to my sales page and all the items uploaded by the current user will be displayed                               | Redirect to my sales page and all the items uploaded by the current user will be displayed                               | Yes |
| P13 | Click on any item in my sales page  | Details of the corresponding item and a "Sold" button will be displayed  | Details of the corresponding item and a "Sold" button will be displayed  | Yes |
| P14 | Click the "Sold" button in the item page  | A confirm message will appear  | A confirm message will appear  | Yes |
| P15 | Click yes for the confirm message   | A "Sold" stamp will be marked on the corresponding item and when any user visit that item will be told the item was sold | A "Sold" stamp will be marked on the corresponding item and when any user visit that item will be told the item was sold | Yes |
| P16 | Click no for the confirm message  | Return to the item page  | Return to the item page  | Yes |
| P17 | Click the "Edit profile" button   | Redirect to the edit profile page  | Redirect to the edit profile page  | Yes |
| P18 | Make change on any personal info. in the edit profile page and click confirm button | Redirect to "My profile" and the contact info. will be edited  | Redirect to "My profile" and the contact info. will be edited  | Yes |
| P19 | Click the "Pay by Alipay" button  | Start and redirect to the Alipay application   | Start and redirect to the Alipay application   | Yes |
| P20 | Click the "Get by Alipay" button  | Start and redirect to the Alipay application   | Start and redirect to the Alipay application   | Yes |
| P21 | Click the "Shake" button  | Redirect to the shake page   | Redirect to the shake page   | Yes |

User profile

| ID | Input   | Expected result  | Actual result  | Result |
|----|---|--|--|--------|
| I1 | Wait for few second with any item has two or more photo                               | The photo display fragment will automatically display other images | The photo display fragment will automatically display other images | Yes    |
| I2 | Slide left or right on the image display fragment with any item has two or more photo | It will switch between different images                            | It will switch between different images                            | Yes    |
| I3 | Click the heart shape button next to the price  | The corresponding item will be added to                            | The corresponding item will be added to                            | Yes    |

|    |   |  |  |     |
|----|---|--|--|-----|
|    | when it is hollow   | the favourite list of the current user and the heart will become solid | the favourite list of the current user and the heart will become solid |     |
| I4 | Slide to the bottom of the page and click the call button | Redirected to the telephone with the prefilled seller phone number     | Redirected to the telephone with the prefilled seller phone number     | Yes |

Item

| ID  | Input                  | Expected result  | Actual result  | Result |
|-----|------------------------|--|--|--------|
| SH1 | Do nothing             | Nothing will happen  | Nothing will happen  | Yes    |
| SH2 | Shake the mobile phone | A sound effect will be played twice with animation than will be redirected to a random items detail page | A sound effect will be played twice with animation than will be redirected to a random items detail page | Yes    |

Shake

## 4. Unfinished functions and future enhancement

### 4.1 Unfinished functions

We have some unfinished function such as comment and user authentication due to the limitation of time but we already have the idea how to finish these two function.

#### User authentication:

After user register a new account, the email is limited in the format xxxx@connect.ust.hk.

The detail of the user information will be store in the database, and the asp application in the server will immediately send an authentication email to this email address with the random authentication code. User authentication is done after user clicks the URL in the email.

#### Comment:

The reason why we did not develop this function is that we consider comment is not that necessary for a second-hand exchange app. In an exchange process, there are only one buyer and seller, so no one else will give comment to this item except the only one buyer.

The previous development scheme is as below:

| name                  | type            | Isnull | 默认值              |
|-----------------------|-----------------|--------|------------------|
| <b>Commentid (PK)</b> | int(11)         | no     | <auto_increment> |
| userid                | int(11)         | yes    | <null>           |
| itemid                | int(11)         | yes    | <null>           |
| content               | text            | yes    |                  |
| createtime            | timestamp       | yes    | <null>           |
| isEnabled             | int(4) unsigned | yes    | 1                |
| sort                  | int(11)         | yes    | <null>           |

- First, establish a new table in the database to store the comment content.
- Establish a new activity associated to the item Activity. It mean we will add a button near by the call button in the Activity.
- The new activity named commentActivity will include a listview. We will develop an adapter for this listview in the same way as any other activity in our project.
- Override an asynctask to call the web method which will download the comment content from the database (through the asp application). Then fulfill the listview with the content.
- Similarly, override an asynctask to call the web method which will upload the comment content. Parameters in the method include content, userid, itemid, createtime.

## 4.2 Future enhancement

We also have some ideas for the future improvements. First of all, graphic is one of the most important feature in this apps, although we have referred the FRESO library to handle the image, we will make full use of this library in the future.

- Specify several different uris for an image, and choose the one already in cache for display
- Show a low-resolution image first and swap to a higher-res one when it arrives
- If the image has an EXIF thumbnail, show it first until the full image loads (local images only)
- Resize or rotate the image
- Download and save image into local storage cache.
- Improve the component related to image.

Moreover, some new features can be added into S-Trade such as instant message for user communication, locate user for meeting between buyer, notification for the favourite items support and also supporting more contact method by using different communication applications like whatsapp and telegram etc. We hope we can try more on S-Trade to enhance ourselves on writing android applications.

## **5. Difficulties encountered**

We have encountered some difficulties when we building S-Trade. First of all, none of us have java develop experience before. Therefore, we need to spend some time to be familiar with the basic syntax of java. Second, it is hard to cooperate with others to build the same function. Since all of us have different programming style and we do not have consensus in how to build some activities. Sometimes we got the semi-finish code from others. Most of the time are wasted to asking what had been done by others and why he do it by that way. So, at the beginning of the project our progress is slow. After that we tried to come up with a consensus of how the design is before we start building the function. The progress become faster.

## **6. Conclusions**

S-Trade is hoped to be a user-friendly and convenience application for HKUST students and staff. It is expected to reduce the waste and encourage reusing. Most of the functions and features we planned to include in S-Trade are finished. We are satisfied with the final product we done and we do hope that it can be used in the future. We also learn a lot from the course and during working on the project. It is a very good course and experience. It has been a pleasure for us to work with each other and learn from Prof. MUPPALA and Mr. LO. We hope we can meet at the future course or study.

## **7. References**

- |                          |   |
|--------------------------|---|
| WeiChatShake.            | <a href="https://github.com/changer0/WeiChatShake">https://github.com/changer0/WeiChatShake</a>                                     |
| multiple-images-selector | <a href="https://github.com/zfdang/android-multiple-images-selector">https://github.com/zfdang/android-multiple-images-selector</a> |
| ViewPager                | <a href="https://github.com/Jude95/ViewPager">https://github.com/Jude95/ViewPager</a>   |